

DOUBLE DATA RATE SYNCHRONOUS SRAM WITH 100% BUS UTILIZATION

5

Stanley A. Hronik

BACKGROUND

Field of the invention

10 The present invention relates to memory circuits, and more particularly to synchronous static random access memories (SRAMs) capable of transferring two data items per clock cycle and 100% bus utilization.

Description of related art

15 Asynchronous SRAMs have no input or output registers. Accessing an asynchronous SRAM is slow because the address and control signals (and the write data in case of a write operation) presented to the SRAM can not be changed for the duration of the SRAM access.

20 Synchronous SRAMs eliminate the requirement to hold the SRAM input signals (address, control, and data) during read or write operations by including clocked registers for storing the address, control, and read and write data. The set-up and hold times for the registers are typically much shorter than the time to access the memory array of the SRAM. This significantly reduces the SRAM's cycle time as viewed at the pins of the device, and thus the frequency of the system clock is increased.

25 *Sub* The input and output registers however, cause two clock cycles of latency *al* in the relation between the read address and read data, and no latency between the write address and write data (i.e., address is clocked in and data is clocked out in two consecutive clock cycles for a read, and the address and data are clocked in in the same clock cycle for a write). This latency difference between read and write

AM
cont. } operations causes the address bus to remain idle for two clock cycles when a read cycle is followed by a write cycle, and causes the data bus to remain idle for two clock cycles when a write cycle is followed by a read cycle (i.e., bus turnaround). The idle cycles reduce the system data bandwidth.

5 Late write SRAMs partially correct the latency problem. In a late write SRAM, the number of idle cycles in a bus turnaround is reduced from two clock cycles to one by introducing one clock cycle of latency in the write. Zero bus turnaround (ZBT) synchronous SRAMs developed by Integrated Device Technology Inc. (Patent Number 5,828,606, issued October 27, 1998) eliminate
10 idle cycles in a bus turn around by causing read and write operations to have the same clock cycle latency of two, and thus achieve 100% bus utilization. The two clock cycles of latency are however undesirable. Fewer cycles of latency, e.g., one, provide faster data availability and potentially faster and easier system design. (The ZBT SRAM latency can be reduced to one clock cycle, but only if no
15 registers are provided on the SRAM output. This is undesirable because it increases the minimum cycle time.)

In all of the above SRAMs, at most one data item is transferred per clock cycle in either a read or a write operation. Double data rate (DDR) SRAMs transfer data into or out of the device on both the rising and falling clock edges,
20 thus doubling the data transfer rate without increasing the clock frequency. One such device is the DDR late-write SRAM known as Claymore or MSUG-2 developed by a private consortium known as the Motorola Semiconductor Users Group (MSUG). This device was designed for high performance workstation level 2 cache operating in a point-to-point environment with data rates in excess of
25 500MHz. While the Claymore device meets the needs for increased bandwidth in high performance communications applications, the lost clock cycle in every bus turn around (associated with every late write device) results in inefficient use of the address and data buses.

Thus, there is a need for a synchronous DDR SRAM capable of 100% bus utilization with fewer clock cycles of latency than the ZBT SRAM.

SUMMARY

5 A synchronous memory circuit is capable of double data transfer rate per clock cycle, 100% bus utilization (i.e., no idle clock cycles in bus turn arounds), and has only one clock cycle of latency in each of read and write burst operations.

10 The synchronous memory circuit includes: an address bus for receiving an address; at least two memory blocks; and a data bus for receiving a data item for transfer to or from the at least two memory blocks, wherein in two consecutive clock cycles at least a first and a second write data items corresponding to a first write burst operation are capable of being transferred to the memory circuit via the data bus and at least a first and second read data items corresponding to a first read burst operation are capable of being transferred from the memory circuit via the data bus.

15 In another embodiment, the at least first and second write data items are provided on the data bus one clock cycle after the first write burst operation is initiated, and the at least first and second read data items are provided on the data bus one clock cycle after the first read burst operation is initiated.

20 In another embodiment, the memory circuit further includes at least one input terminal for receiving at least one read/write control signal for indicating a read burst or a write burst operation, wherein each of the first write burst operation and the first read burst operation is initiated upon a rising edge of a clock cycle by asserting the read/write control signal to indicate a write burst or a read burst operation and providing a burst address at the address bus both prior to the rising
25 edge of the clock cycle.

In another embodiment, the first write burst operation comprises writing at least a third and fourth write data items to respective two memory blocks so that

writing the third write data item overlaps with writing the fourth write data item, the at least third and fourth write data items corresponding to a last write burst operation prior to the first write burst operation. In another embodiment, each of the at least third and fourth write data items is written in one clock cycle, and
5 writing the third write data item overlaps with writing the fourth write data item during half a clock cycle.

In another embodiment, the first read burst operation comprises reading the at least first and second read data items from respective two memory blocks so that reading the first read data item overlaps with reading the second read data item. In
10 another embodiment, each of the at least first and second read data items is read in one clock cycle, and reading the first read data item overlaps with reading the second read data item during half a clock cycle.

In another embodiment, the memory circuit further includes an output circuit for receiving the at least first and second read data items from respective at
15 least two memory blocks in the first read burst operation and allowing the at least first read data item to be provided on the data bus half a clock cycle after the first read burst operation is initiated, and allowing the second read data item to be provided on the data bus one clock cycle after the first read burst operation is initiated.

In another embodiment, the output circuit includes a multiplexer for
20 receiving a clock signal and the at least first and second read data items and sequentially transferring to an output bus of the multiplexer the at least first and second read data items in accordance with the state of the clock signal, and an output buffer for receiving the at least first and second read data items from the
25 output bus of the multiplexer and providing the at least first and second read data items to the data bus when enabled, wherein the output buffer is enabled only when a valid read data item is to be provided on the data bus so that no external tracking of the progress of a read burst operation is required.

In another embodiment, the memory circuit is a static random access memory (SRAM).

A method of accessing the synchronous memory circuit includes the acts of: (A) initiating a first write burst operation for sequentially transferring at least a first and second write data items to the memory circuit in a first clock cycle; and
5 (B) initiating a first read burst operation for sequentially transferring at least a first and second read data items from the memory circuit in a second clock cycle, wherein the first and second clock cycles are two consecutive clock cycles.

In another embodiment, the method further comprises: (C) initiating the
10 first write burst operation in a third clock cycle, the first clock cycle being the next sequential clock cycle after the third clock cycle; and (D) initiating the first read burst operation in a fourth clock cycle, the second clock cycle being the next sequential clock cycle after the fourth clock cycle.

In another embodiment, act (C) comprises: (E) asserting a read/write
15 control signal on an input terminal of the memory circuit to indicate a write burst operation prior to a rising edge of the third clock cycle; and (F) providing a first write burst address on an address bus of the memory circuit prior to the rising edge of the third clock cycle. Act (D) comprises: (G) asserting the read/write control signal to indicate a read burst operation prior to a rising edge of the fourth clock
20 cycle; and (H) providing a first read burst address on the address bus prior to the rising edge of the fourth clock cycle.

In another embodiment, the memory circuit includes two memory blocks and act (A) comprises: (I) writing at least a third and fourth write data items to respective two memory blocks so that writing the third write data item overlaps
25 with writing the fourth write data item, the at least third and fourth write data items corresponding to a last write burst operation prior to the first write burst operation. In another embodiment, act (I) comprises: (J) writing the third write data item in one clock cycle; and (K) writing the fourth write data item in one clock cycle,

wherein writing the third write data item overlaps with writing the fourth write data item during half a clock cycle.

In another embodiment, act (B) comprises: (L) reading the at least first and second read data items from respective two memory blocks so that reading the first read data item overlaps with reading the second read data item. In another embodiment, act (L) comprises: (M) reading the first read data item in one clock cycle; and (N) reading the second read data item in one clock cycle, wherein reading the first read data item overlaps with reading the second read data item during half a clock cycle.

Other features and advantages of the invention are described below. The invention is defined by the appended claims.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention may be better understood, and its numerous objects, features, and advantages made apparent to those skilled in the art by referencing the accompanying drawings.

Fig. 1 is a block diagram of a preferred embodiment of a double data rate SRAM in accordance with the present invention.

Fig. 2 is a sample timing diagram showing the waveforms for some signals in the Fig. 1 block diagram.

The use of the same reference symbols in the drawings indicates similar or identical items.

DESCRIPTION OF PREFERRED EMBODIMENTS

Fig. 1 is a block diagram of a DDR integrated SRAM 10. SRAM 10 is a burst synchronous SRAM capable of 100% bus utilization (e.g., no idle cycles in a

bus turn around) with only one clock cycle latency between address and data for both read and write operations.

SRAM 10 has an address bus 201 for receiving an address, a data bus 202 for providing read data and receiving write data, three input terminals 141, 142, and 203 for receiving the respective control signals \overline{CS} , \overline{WE} , and \overline{OE} , and an input terminal 200 for receiving the system clock CLK. SRAM 10 includes two identical memory blocks 20 and 30, five clocked registers 40, 50, 60, 70, and 110, two multiplexers 80 and 120, a comparator 90, a 3-state output buffer 130, and four logic gates 67, 73, 100, and 140. The five clocked registers are clocked by the rising edges of the clock signal at their respective CK input terminals. However, the invention is not limited to the registers being clocked by the rising edges, or by any other particular features or circuitry, except as defined by the appended claims.

OR gate 140 is a two input gate with its first and second input terminals connected to \overline{CS} terminal 141 and \overline{WE} terminal 142 respectively, and its output terminal connected to lead 143. Register 110 has: an input bus I1 connected to address bus 201, and a corresponding output bus Q1; an inverting input terminal CE connected to lead 143; and an input terminal CK connected to CLK terminal 200. Comparator 90 has a first input bus 91 connected to output bus Q1 of register 110, a second input bus 92 connected to address bus 201, and an output terminal 93. Mux 80 is a two to one mux and has a first input bus 81 connected to address bus 201, a second input bus 82 connected to output bus Q1 of register 110, a control input terminal 84 connected to lead 143, and an output bus 83. Mux 80 selects the data on one of its two input buses for transfer to its output bus in accordance with the following table:

Control input 84	Output (bus 83)
-----	-----
0	Address at bus 82
1	Address at bus 81

5

Table 1

Register 40 has: an input terminal I1 connected to \overline{CS} terminal 141, and a corresponding output terminal Q1 connected to lead 43; an input terminal I2 connected to lead 143, and a corresponding output terminal Q2 connected to lead 44; an input terminal I3 connected to output terminal 93 of comparator 90, and a corresponding output terminal Q3 connected to lead 45; an input bus I4 connected to output bus 83 of mux 80, and a corresponding output bus Q4 connected to bus 46; and an input terminal CK connected to CLK terminal 200. Register 50 has: an input bus I1 connected to data bus 202, and a corresponding output bus Q1; an inverting input terminal CE connected to lead 44, and an input terminal CK connected to CLK terminal 200. Inverter 100 has an input terminal connected to CLK terminal 200, and an output terminal connected to lead 102.

Register 60 has: an input terminal I1 connected to lead 43, and a corresponding inverting output terminal Q1; an input terminal I2 connected to lead 45, and a corresponding output terminal Q2; an input terminal I3 connected to lead 44, and a corresponding output terminal Q3 connected to lead 71; an input bus I4 connected to bus 46, and a corresponding output bus Q4; and an input terminal CK connected to lead 102. Register 70 has: an input bus I1 connected to data bus 202, and a corresponding output bus Q1; an inverting input terminal CE connected to the output terminal Q3 of register 60, and an input terminal CK connected to lead 102.

Mux 120 is a four to one multiplexer and has a first input bus 121, a second input bus 122 connected to output bus Q1 of register 50, a third input bus 123, a fourth input bus connected to output bus Q1 of register 70, a first control input

terminal 125 connected output terminal Q2 of register 60, a second control input terminal 126 connected to CLK terminal 200, and an output bus connected to bus 127. Mux 120 selects the data on one of its four input buses for transfer to its output bus in accordance with the following table:

5	Control input	Control input	Output (bus 127)
	126 (CLK)	125	
	0	0	Dout 24
	0	1	Din 23
10	1	0	Dout 34
	1	1	Din 33

Table 2

Output buffer 130 has an input bus connected to bus 127, a control input terminal connected to lead 74, and an output bus connected to data bus 202. AND gate 67 is a two input gate, and has an inverting input terminal connected to $\overline{\text{OE}}$ terminal 203, a second input terminal connected to lead 54, and an output terminal connected to lead 74. AND gate 73 is a two input gate with its first and second input terminals respectively connected to inverting output terminal Q1 and output terminal Q3 of register 60, and its output terminal connected to lead 54.

Gate 300 is a two input, two output AND gate, and has a first input terminal connected to lead 54, a second input terminal connected to CLK terminal 200, a first output terminal 301 for providing EchoCLK signal and a second inverting output terminal 302 for providing $\overline{\text{EchoCLK}}$ signal. Gate 300 also has a control input terminal connected to $\overline{\text{OE}}$ terminal 203.

Memory blocks 20 and 30 are internally identical, each having a read/write control input terminal $\overline{\text{W}}$, an address bus Add, a data-in bus Din, and a data-out bus Dout. Memory block 20 has: $\overline{\text{W}}$ terminal 21 connected to lead 44, Add bus 22

connected to bus 46, Din bus 23 connected to output bus Q1 of register 50, and
 Dout bus 24 connected to input bus 121 of mux 120. Memory block 30 has: \overline{W}
 terminal 31 connected to output terminal Q3 of register 60, Add bus 32 connected
 to output bus Q4 of register 60, Din bus 33 connected to output bus Q1 of register
 5 70, and Dout bus 34 connected to input bus 123 of mux 120. Memory blocks 20
 and 30 operate asynchronously. If the \overline{W} input of block 20 or 30 is low, the block
 stores the data provided on its input Din. If \overline{W} is high, the block provides read
 data on output Dout.

Fig. 2 is a sample timing diagram showing the waveforms for CLK, \overline{CS} ,
 10 \overline{WE} , address, data, EchoCLK, $\overline{\text{EchoCLK}}$ and some of the internal leads of SRAM
 10. Twelve clock cycles (N to N+11) are illustrated. Each clock cycle starts at the
 rising edge of system clock CLK. Shaded areas indicate the times that the signal is
 allowed to change. The number next to each signal name corresponds to the
 numbered buses, terminals, and leads in Fig. 1.

15
 Data is transferred to or from SRAM 10 in bursts of two data items. The
 two data items are both either read or write data. Each burst is initiated on the
 rising edge of clock CLK. In a read burst, the address is provided to memory block
 20 20 (through register 40) on the rising edge starting the burst operation. For
 example, in the read started on the rising edge of cycle N+3 in Fig. 2, the read
 address A6 is provided to block 20 on the rising edge of N+3. The same address is
 provided to memory block 30 (through register 60) on the next falling edge.
 Memory blocks 20 and 30 provide respective data on their respective Dout buses.
 Multiplexer 120 selects the Dout bus from block 20 when clock CLK becomes low
 25 (i.e. starting the falling edge), and selects the Dout output from memory block 30
 when the system clock becomes high (starting the rising edge of the next clock
 cycle).

In a write burst, the address is latched by SRAM 10 before the data. For example, in the burst initiated in clock cycle N, the write address A0 is latched by register 110 from bus 201 at the rising edge of cycle N, the first data item D0w is latched by register 50 from bus 202 at the rising edge of cycle N+1, and the second data item D1w is latched by register 70 from bus 202 at the falling edge of cycle N+1. However, the write address is delayed by register 40, so the registers 40 and 50 supply the write address and the first data item to memory block 20 at the same time. Registers 60 and 70 supply the write address and the second data item to memory block 30 at the same time.

If the write operation is immediately followed by a read or a dead cycle, the write data are not written to memory blocks 20 and 30. The write data are stored in registers 50 and 70, and the write address is stored in register 110, until the next write operation.

We now describe the write, read sequence on the example of the A12 write, A14 read bursts in more detail. The write burst at address A12 is initiated at the rising edge of clock cycle N+6 by asserting a low voltage on the \overline{WE} terminal 142 (\overline{CS} signal must be active, i.e., a low voltage). A burst of two data items D12w and D13w corresponding to the external address A12 are provided at data bus 202 at the respective rising and falling edges of clock cycle N+7. Signal R/W at the output of OR gate 140 (which is the same as the \overline{WE} signal when \overline{CS} is active) is used to set up registers 110, 50, and 70 for storing the respective address A12 and data D12w and D13w by pulling their inverted CE input terminals low at the proper time. The CE terminal of register 110 is pulled low just prior to the N+6 cycle. The CE terminal of register 50 is pulled low through register 40 upon the rising edge of clock cycle N+6. The CE terminal of register 70 is pulled low through registers 40 and 60 upon the falling edge of clock cycle N+6.

Thus, address A12 is stored in register 110 at the rising edge of clock cycle N+6, and data D12w and D13w are stored in the respective registers 50 and 70 at

the respective rising and falling edges of the next clock cycle N+7. The address and data are maintained in these registers until the next write cycle, i.e., cycle N+10. At the rising edge of clock cycle N+10, the low \overline{WE} signal initiating the write burst at address A20 is provided to the \overline{W} terminal of memory block 20 through OR gate 140 and register 40. This causes data D12w in register 50 to be written to memory block 20 at the internal address A12. Half a clock cycle later, at the falling edge of clock cycle N+10, \overline{WE} low is provided to the \overline{W} terminal of memory block 30 through register 60. This causes data D13w in register 70 to be written to memory block 30 at the internal address A12.

Note that address A12 stored in register 110 is selected by mux 80 just prior to clock cycle N+10 when the R/W signal is low (in accordance with Table 1), and is provided to the Add bus 22 of memory block 20 through register 40 at the rising edge of clock cycle N+10. Half a clock cycle later, at the falling edge of clock cycle N+10, address A12 is provided to the Add bus 32 of memory block 30 through register 60.

Also note that the \overline{WE} low signal initiating the A12 write burst in the N+6 clock cycle also causes data D8w and D9w (stored in respective registers 50 and 70 during the preceding write burst operation) to be written to respective memory blocks 20 and 30 at the internal address A8 (stored in registers 110 and 60 during the preceding write burst operation).

The read burst at address A14 is initiated at the rising edge of clock cycle N+7 by asserting a high voltage on the \overline{WE} terminal 142 (the \overline{CS} signal must be active). Two data items D14r and D15r corresponding to the external address A14 are sequentially provided on data bus 202 at the respective rising and falling edges of clock cycle N+8. With a high \overline{WE} signal, registers 110, 50, and 70 are disabled through their respective CE input terminals at the proper time. The high \overline{WE}

signal propagates through OR gate 140 and causes mux 80 to select address A14 on address bus 201 in accordance with Table 1.

Upon the rising edge of clock cycle N+7, address A14 and the high \overline{WE} signal are clocked in by register 40 and presented respectively to Add bus 22 and \overline{W} terminal 21 of memory block 20. Register 40 holds A14 and the high \overline{WE} signal for the full clock cycle N+7. Memory block 20 provides data D14r, which corresponds to the internal address A14, to input bus 121 of mux 120 via Dout bus 24. In accordance with Table 2, mux 120 selects its input bus 121 when clock CLK becomes low (i.e., data D14r can pass through mux 120 on or after the falling edge of clock cycle N+7). 3-state output buffer 130 provides the data on its input bus 127 to data bus 202 when the signal at its 3-state input terminal 74 goes high. Thus, data D14r is provided on data bus 202 in the second half of clock cycle N+7 as shown in Fig. 2 (note that the \overline{OE} signal must be active).

The address A14 and the high \overline{WE} signal are passed on to memory block 30 by register 60 upon the falling edge of clock cycle N+7. Register 60 holds A14 and the high \overline{WE} signal for a full clock cycle. Memory block 30 provides data D15r, which corresponds to the internal address A14, to input bus 123 of mux 120 via Dout bus 34. In accordance with Table 2, mux 120 selects its input bus 123 when clock CLK becomes high (i.e., data D15r can pass through mux 120 on or after the rising edge of clock cycle N+8). Output buffer 130 then provides data D15r to data bus 202 since its 3-state signal is high, as shown in Fig. 2.

The 3-state signal of output buffer 130 is controlled by the \overline{WE} signal through registers 40 and 60, and AND gates 73 and 67. The 3-state signal is used to ensure that output buffer 130 is enabled only for read bursts. As indicated in Fig. 2, the 3-state signal is active only if and whenever valid read data is provided on data bus 202. This allows the initiation of a read burst without requiring external tracking of the progress of the burst.

In a read burst, each read data item can be made available on data bus 202 for an entire half cycle if the address-to-data time of memory blocks 20 and 30 is a half cycle or less. For example, data item D6r can be provided on data bus 202 starting at the falling edge of cycle N+3, and not later as in Fig. 2. However, for a given address-to-data time of memory blocks 20 and 30, the ability to provide the read data item on data bus 202 more than a half cycle after the initiation of the read burst enables the clock frequency to be increased compared to the case wherein the read data item is required to be available on data bus 202 within a half cycle after the initiation of the read burst.

Because of the use of mux 120 instead of an output register, the time from clock CLK to data out is similar to a registered output, but the access of memory blocks 20, 30 is asynchronous. This enables a faster memory access time by not requiring the data setup time prior to clock of an output register since the mux selection can be made as the data is propagating through device 10. Also, by eliminating the output register, the latency between read address and read data is reduced from two clock cycles to one. Mux 120 may be replaced by a controlled selection device, for example, a driver which can be enabled at the appropriate time in the cycle.

Note that the advantages of using mux 120 can also be realized in conventional synchronous memory devices (e.g., synchronous DRAMs and PROMs) by using a clock-controlled selection circuit instead of the conventional output register. The selection circuit similar to mux 120 allows read data to pass through to the output bus if the clock condition is met. In other words, the selection circuit is enabled by the clock signal half a clock cycle after the read address is clocked in by the input register (e.g., register 40). The selection circuit may be a simple transmission gate controlled by the clock signal.

As the above description of the A12 read burst, A14 write burst sequence indicates, having two separate memory blocks 20, 30 allows a read from one

memory block to overlap with a write to the other memory block. For example, in the first half of cycle N+7, the data item D13w is being written to memory block 30 while the data item D14r is being read from memory block 20.

A special condition arises when the read address equals the latest preceding write address. Under this condition, the latest preceding write data have not yet
 5 been written to the memory blocks 20 and 30, but a read request has been issued for the data. The read is handled as follows.

At the start of the latest preceding write burst, the write address was written to register 110 as described above. After that, no writes have taken place, so the
 10 $\overline{\text{CS}}$ signal or the $\overline{\text{WE}}$ signal, or both, have been high on the rising CLK edges. Hence, the CE input of register 110 has been high on the rising CLK edges, so register 110 has continued to store the latest preceding write address. Comparator 90 always compares an incoming address received at its input bus 92 with the write address stored in register 110. If the two addresses match, comparator 90, through
 15 registers 40 and 60, causes the control input terminal 125 (marked as "=") of mux 120 to go high at the appropriate time. In accordance with Table 2, if clock CLK is low, mux 120 selects the data provided by register 50 on input bus 122 rather than the data provided by memory block 20 on input bus 121. If clock CLK is high, mux 120 selects the data provided by register 70 on input bus 124 rather than the
 20 data provided by memory block 30 on input bus 123.

Once a burst (read or write) is initiated, it progresses to completion. Thus, control signals $\overline{\text{WE}}$ and $\overline{\text{CS}}$ and the address need not be maintained throughout the burst. If dead cycles are required, they can be inserted using the $\overline{\text{CS}}$ and/or $\overline{\text{OE}}$ control signals.

25 The burst length can be made greater than two by providing a corresponding greater number of internal memory blocks. As with the length of two, longer bursts are carried out as the same internal address is passed from one

memory block to the next on each edge of the clock. A longer burst causes the data bus to be occupied for more clock cycles, and allows the address bus and the control signal \overline{WE} to be free for longer periods.

The data rate on SRAM data bus 202 is twice the data rate on any one of the Dout buses of memory blocks 20 and 30. One reason for this is that the reading from memory blocks 20 and 30 overlaps. A full clock cycle is available for each read from each memory block. For example, in a read from memory block 20 in the A6 read burst, A6 is provided to block 20 at the start of clock cycle N+3, and the data D6r can be read out to data bus 202 any time before the end of cycle N+3, provided the set-up and holding times for the reading (target) device (not shown) are satisfied. The block 30 read is also allowed to take one full cycle starting at the falling edge of cycle N+3. The two reads overlap in the second half of cycle N+3.

Similarly, the writes to blocks 20 and 30 overlap. Although the actual write to memory blocks 20 and 30 occur in a write cycle other than the one to which the write data corresponds, nevertheless, registers 50 and 70 are clocked at half the speed at which the data are provided on data bus 202. A whole cycle is allowed to write to a memory block, and one half a cycle is allowed for overlap.

Further, in a write burst following a read burst, a write to block 20 and a read from block 30 overlap. Similarly, if a read burst follows a write burst, a read from block 20 and a write to block 30 overlap.

Hence, the external address on bus 201 is at half the toggle rate of the clock CLK for a burst of two. If longer burst lengths are implemented, the external address could be even slower.

SRAM 10 can be disabled during any clock cycle by holding \overline{CS} high on the rising edge of the clock. While disabled, SRAM 10 completes any activities initiated in previous clock cycles. Internally, SRAM 10 performs a read operation on whatever address is on address bus 201 at the initiation of the disable cycle.

This is because when \overline{CS} is high, the R/W signal on lead 143 goes high (i.e., a read operation). But, no data will be driven to data bus 202 because the high \overline{CS} signal propagates through registers 40 and 50, and AND gates 73 and 67, and disables buffer 130 prior to the time the data for the disabled cycle reaches output
 5 buffer 130.

SRAM 10 can be configured to have a dedicated data-in bus for write data and a dedicated data-out bus for read data by simply connecting the I1 input bus of registers 50 and 70 to the data-in bus, and the output bus of output buffer 130 to the data-out bus. With separate data buses, there is no bus turned around and thus
 10 there are no data contention issues. However, in a read cycle the data-in bus is idle, and in a write cycle the data-out bus is idle.

Typically, in a read operation the clock along with the read instruction are delivered to the SRAM and the SRAM returns the indicated response to a target device. In very high speed clocking arrangements and where the target device is a
 15 clocked device, coordinating the clock with the return data is difficult because of latencies and differences in routing of the data and the clock. The result is that the data read from the SRAM is not coordinated with the clock (the data often arrives at the target device later than that allotted by the clock cycle), and read errors at the target device occur.

To eliminate the potential read errors at the target device, dedicated clock signals EchoCLK and $\overline{EchoCLK}$ are provided for being routed to the target device along with the read data. By ensuring that the EchoCLK and $\overline{EchoCLK}$ timing coincide with the read data availability at the data bus 202, any potential timing
 20 skews between the clock and the data at the target device reading the SRAM 10 are eliminated. Further, the EchoCLK and $\overline{EchoCLK}$ signals are active only during read bursts (i.e., these signals change with clock CLK only when the 3-state signal is high). This simplifies the system design by eliminating the need for a separate signal notifying the target device of the data transfer from SRAM 10.

As shown in Fig. 1, EchoCLK / $\overline{\text{EchoCLK}}$ signals are generated by AND gate 300. Gate 300 allows EchoCLK / $\overline{\text{EchoCLK}}$ signals to respond to the CLK signal only when read data is provided on data bus 202. EchoCLK / $\overline{\text{EchoCLK}}$ signals do not respond to the CLK signal when either the $\overline{\text{OE}}$ control signal is high, or when lead 54 is low (i.e. when data bus 202 is in 3-state). Gate 300 should be designed to have a gate delay shorter than or equal to that of mux 120 plus output buffer 130. This ensures that EchoCLK can be used externally with a zero hold time register (not shown) to capture any data appearing on data bus 202. The EchoCLK / $\overline{\text{EchoCLK}}$ signals help the system achieve the maximum possible setup time on the target device, while the hold time is minimized.

The $\overline{\text{EchoCLK}}$ signal is provided to facilitate clocking of data on both the rising and falling edges of clock CLK. The rising edge of EchoCLK can be used to clock rising edge data, and the rising edge of $\overline{\text{EchoCLK}}$ can be used to clock falling edge data. Also, EchoCLK and $\overline{\text{EchoCLK}}$ can be used differentially if the application requires it. Note that the EchoCLK and $\overline{\text{EchoCLK}}$ signals are not necessary for the proper operation of SRAM 10, and merely support the external use of the SRAM. Thus, the EchoCLK and $\overline{\text{EchoCLK}}$ signals and the associated circuitry may be eliminated if the application does not use them.

In accordance with the invention, among other features and advantages, double data transfer rate per clock cycle, only one clock cycle of latency in each of read and write bursts, and 100% bus utilization (i.e., no idle clock cycles in bus turn arounds) are achieved.

The dual data rate SRAM of the present invention is intended for but not limited to high speed read/write applications, such as network switches and routers, which receive and store data in a memory before data are transmitted (here both read and write can be carried out without interfering with each other), or graphics

applications where data is loaded into the graphics memory and then fed out continuously to a video screen.

The above description of the present invention is intended to be illustrative and not limiting. The invention is not limited to any particular circuitry or timing, to the number of external address bits provided to the memory, to any signal being
5 provided on a rising or falling edge, or to edge-sensitive circuitry. The invention is not limited to an integrated SRAM 10, i.e., discrete components may be used to implement SRAM 10. The invention is not limited to any particular type of memory blocks 20 and 30, which can be asynchronous, synchronous, or perhaps
10 other types of memories, known or to be invented. The invention includes all variations and modifications falling within the scope of the appended claims.